UPS: Unified Parallel Software

Large scale physics simulations must be designed to execute effectively in a distributed memory, parallel processing environment. UPS, an acronym for “Unified Parallel Software”, is a library of routines designed to help the application developer create efficient, portable, extensible, and robust large scale parallel programs for such applications. Some parallel programming models attempt to hide the parallelism from the application writer. Others require that the application writer work at the lowest levels. UPS falls in between: it is designed to expose the parallel environment to the computation while abstracting away the necessary complexities. The result is a simplified coding style, natural to the application, which minimizes the time spent moving data among the distributed processes.

UPS is designed to run in any computing environment that supports the C programming language and which provides a method for moving data between parallel processes, such as MPI. However, the higher level interface between the application and the computing environment allows for performance optimizations based on available hardware and software characteristics and components.

More on UPS

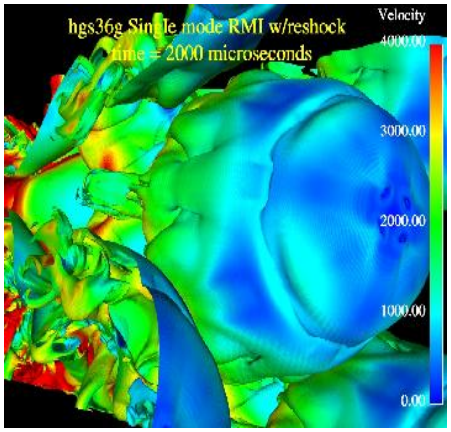
UPS is composed of packages or components, each of which is designed to address a category of tasks common to writing parallel applications. For example, the communication component includes reductions and broadcasts. The “meat” of UPS is on the movement of data between distributed processes. Special functionality is included which can help keep track of complex communication requirements, such as is needed by, for example, physics applications which operate on unstructured meshes.

Our users asked us to provide a light-weight infrastructure, one that required no changes to the data structures they had already developed and invested in. Our users also rely heavily on the close coupling of the UPS team to their projects so that we may quickly respond to their particular needs. In this regard, UPS can be viewed as a “blue-collar” alternative to existing parallel infrastructures.

This figure shows the result of a RAGE simulation of a single-mode single-interface (SF6-air) Richtmyer-Meshkov Instability (Mach 1.2 shock) that has been shocked at time = 0 and then later re-shocked (time ~ 0.9 msec) by the reflected shock.

Gather/scatter

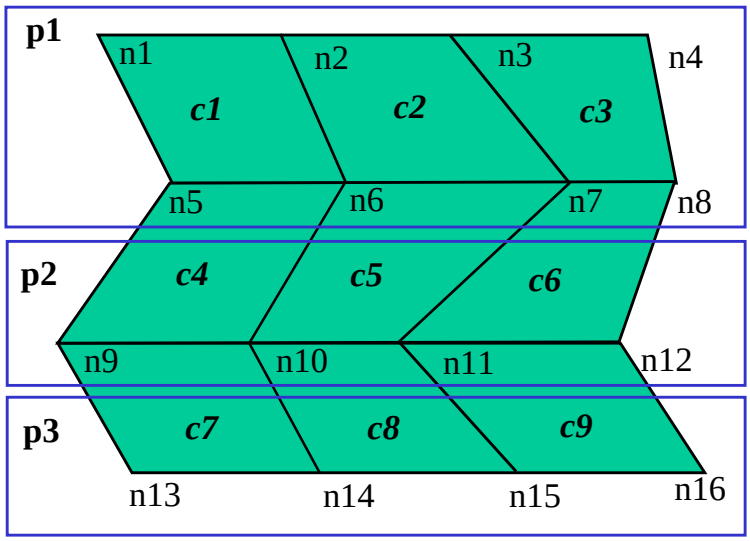
Gather/scatter functions let the application submit a request for data to be collected into a specified address in memory. UPS collects the data from memory addresses that are both local to the process and located in the memory owned by other



processes. This abstracts a significant amount of work away from the application writer, and allows the developers of UPS to optimize the movement of data.

The following is an example of how users might apply the gather/scatter component.

The figure illustrates a domain decomposition and cell connectivity scheme common to applications that could benefit from using the UPS



gather/scatter component. The domain consists of nine cells with sixteen associated vertices, or nodes. The nodes are decomposed into three partitions as follows:

Partition P1 owns nodes n1-n8 and cell centers c1-c3. Partition P2 owns nodes n9-n12 and cell centers c4-c6. Partition P3 owns nodes n13-n16 and cell centers c7-c9

So, if process P2 wishes to get the values for all the cell centers (both on and off process) it can make a single gather call. Later on, the process can reverse the process and scatter information back to the cell centers with a single call to scatter.

Collaborative Effort

UPS relies heavily on incremental contributions from a wide variety of sources. Success can only be assured through the cooperative efforts of the UPS core team and all of its contributors. As such, UPS is designed to provide an efficient means for getting the work of software developers, researchers, and other interested parties incorporated into user applications. Further, UPS provides an excellent opportunity for helping hardware, firmware, and software providers understand how a significant portion of ASCI applications make use of a computing environment.

Our current list of collaborators come from divisions throughout the laboratory (CIC, MST, X), universities (Tennessee, Minnesota), partner laboratories (Lawrence Livermore, Oak Ridge, Sandia-Albuquerque, and Sandia-California), and third party vendors (Abba Technologies, CPCA, Intel, and SAIC). In preparation for execution on other compute platforms, we have interacted with employees from Sun Microsystems, Compac, HP, and IBM. Interaction with our collaborators has resulted in significant performance improvements on ASCI Blue Mountain. Our goal is to continue making such improvement on this and other platforms.

For further information, contact Richard Barrett at *rbarrett@lanl.gov* or Mike McKay, Jr. at *lmdm@lanl.gov*. The UPS web page is found at

www-xdiv.lanl.gov/XCI/PROJECTS/UPS